IN THE CLAIMS:

Please amend the claims as follows:

1. (previously presented) A re-multiplexer module comprising:

an input processor controlled by a host processor in a packet processing system;

a packet buffer;

a packet identifier table, said table comprising an active table containing values used by the input processor to select packets for storage in said packet buffer; and a pending table containing values that can be modified by the host processor while the active table is being used by the input processor,

wherein said input processor stores a packet in said packet buffer if said packet has an identifier listed in said packet identifier table.

2. (previously presented) The re-multiplexer module of claim 1, wherein the packet identifier table is constructed as a multi-port accessible memory.

3. (previously presented) The re-multiplexer module of claim 2, wherein the multi-port accessible memory is a dual-port accessible memory.

4. (previously presented) The re-multiplexer module of claim 3, wherein the dual-port accessible memory is partitioned into two portions to that respectively hold the active table and the pending table.

5. (previously presented) The re-multiplexer module of claim 1, further including a switching mechanism allowing the host processor to make the pending table the active table, such that the active table becomes the pending table.

6. (previously presented) The re-multiplexer module of claim 5, wherein the switching mechanism includes modifying a control bit in the input processor via the host processor.

7-9. (cancelled)

10. (previously presented) The re-multiplexer module of claim 1, further comprising control logic that detects a length of a packet being written to said packet buffer, wherein, if said packet length is incorrect, said input processor will overwrite that packet in said packet buffer.

11. (previously presented) The re-multiplexer module of claim 1, further comprising control logic that detects a transport error in a packet, wherein, if a transport error is detected, said input processor will disable writing of that packet to said packet buffer.

12. (previously presented) The re-multiplexer module of claim 1, further comprising a time reference generator, wherein said input processor associates a time stamp with each packet.

13. (previously presented) The re-multiplexer module of claim 12, wherein said time stamp is generated for a particular packet when a last byte of that packet is received by said input processor.

14. (previously presented) The re-multiplexer module of claim 12, further comprising an output processor that reads packets from said packet buffer and selectively adds said packets to one of two or more output data streams, wherein said output processor uses said time stamp to calculate an amount of time taken to pass a particular packet through said re-multiplexer module.

15. (previously presented) The re-multiplexer module of claim 14, wherein said output processor further comprises Program Clock Reference (PCR) correction circuitry that removes PCR jitter using said amount of time taken to pass a particular packet through said re-multiplexer module.

16. (previously presented) The re-multiplexer module of claim 1, wherein said input processor further comprising a Program Clock Reference (PCR) detector that flags packets containing PCR data.

17. (previously presented) The re-multiplexer module of claim 16, further comprising PCR correction circuitry in an output processor that extracts PCR data from said packets flagged as containing PCR data.

18. (previously presented) The re-multiplexer module of claim 1, further comprising an output processor that reads packets from said packet buffer and selectively adds said packets to one of two or more output data streams.

19. (previously presented) The re-multiplexer module of claim 18, wherein said output processor assigns a new packet identification (PID) number to packets in said two or more output data streams.

20. (previously presented) The re-multiplexer module of claim 1, wherein said packet buffer stores, for each stored packet, an input timestamp, a Packet Identification (PID) number and the packet itself.

21. (previously presented) The re-multiplexer module of claim 20, wherein said packet buffer further stores, for each stored packet, a Program Clock Reference Flag.

22. (previously presented) The re-multiplexer module of claim 1, wherein said packet identifier table further identifies some packets as having priority status such that, if said packet buffer is filled to a predetermined point, said input processor writes only packet having priority status to said packet buffer.

23. (previously presented) The re-multiplexer module of claim 22, wherein said predetermined point is half full.

24. (previously presented) A method of re-multiplexing data packets with a re-multiplexer module, said method comprising:

checking a packet identification (PID) number for each incoming packet against entries in a packet identifier table;

storing a packet in a packet buffer if a PID number for that packet is found in said packet identifier table;

dividing said packet identifier table into an active table and a pending table; and

modifying said pending table while said active table is in use by an input processor performing said checking and storing of data packets.

25. (previously presented) The method of claim 24, further comprising selectively switching a status of each portion of said table such that said active table becomes said pending table and said pending table becomes said active table.

26. (previously presented) The method of claim 24, further comprising:

detecting a length of a packet being written to said packet buffer; and,

if said packet length is incorrect, overwriting that packet in said packet buffer.

27. (previously presented) The method of claim 24, further comprising:

detecting a transport error in a packet; and,

if a transport error is detected, disabling writing of that packet to said packet buffer.

28. (previously presented) The method of claim 24, further comprising associating a time stamp with each packet.

29. (currently amended) The method of ~~claim 12~~ claim 28, further comprising:

reading packets from said packet buffer;

selectively adding said packets to one of two or more output data streams;

using said time stamp to calculate an amount of time taken to pass a particular packet through said re-multiplexer module; and

removing Program Clock Reference (PCR) jitter using said amount of time taken to pass a particular packet through said re-multiplexer module.

30. (previously presented) The method of claim 24, further comprising flagging packets containing Program Clock Reference (PCR) data.

31. (previously presented) The method of claim 30, further comprising extracting PCR data from said packets flagged as containing PCR data.

32. (previously presented) The method of claim 24, further comprising:

reading packets from said packet buffer; and

selectively adding said packets to one of two or more output data streams.

33. (previously presented) The method of claim 32, further comprising assigning a new packet identification (PID) number to packets in said two or more output data streams.

34. (previously presented) The method of claim 24, further comprising identifying some packets in said packet identifier table as having priority status such that, if

said packet buffer is filled to a predetermined point, only packet having priority status are written to said packet buffer.

35. (previously presented) The method of claim 34, wherein said predetermined point is half full.

36. (previously presented) A re-multiplexer module comprising:

an input processor controlled by a host processor in a packet processing system

a packet buffer;

a packet identifier table wherein said input processor stores a packet in said packet buffer if said packet has an identifier listed in said packet identifier table; and

an output processor that reads packets from said packet buffer and selectively adds said packets to one of two or more output data streams.

37. (previously presented) The re-multiplexer module of claim 36, wherein said table comprises:

an active table containing values used by the input processor to select packets for storage in said packet buffer; and

a pending table containing values that can be modified by the host processor while the active table is being used by the input processor,

38. (previously presented) The re-multiplexer module of claim 37, further including a switching mechanism allowing the host processor to make the pending table the active table, such that the active table becomes the pending table.

39. (previously presented) The re-multiplexer module of claim 36, further comprising control logic that detects a length of a packet being written to said packet buffer, wherein, if said packet length is incorrect, said input processor will overwrite that packet in said packet buffer.

40. (previously presented) The re-multiplexer module of claim 36, further comprising control logic that detects a transport error in a packet, wherein, if a transport error is detected, said input processor will disable writing of that packet to said packet buffer.

41. (previously presented) The re-multiplexer module of claim 36, further comprising a time reference generator, wherein said input processor associates a time stamp with each packet.

42. (previously presented) The re-multiplexer module of claim 41, wherein said output processor uses said time stamp to calculate an amount of time taken to pass a particular packet through said re-multiplexer module.

43. (previously presented) The re-multiplexer module of claim 42, wherein said output processor further comprises Program Clock Reference (PCR) correction circuitry that removes PCR jitter using said amount of time taken to pass a particular packet through said re-multiplexer module.

44. (previously presented) The re-multiplexer module of claim 36, wherein said input processor further comprising a Program Clock Reference (PCR) detector that flags packets containing PCR data.

45. (previously presented) The re-multiplexer module of claim 44, further comprising PCR correction circuitry in said output processor that extracts PCR data from said packets flagged as containing PCR data.

46. (previously presented) The re-multiplexer module of claim 36, wherein said output processor assigns a new packet identification (PID) number to packets in said two or more output data streams.

47. (previously presented) The re-multiplexer module of claim 36, wherein said packet buffer stores, for each stored packet, an input timestamp, a Packet Identification (PID) number and the packet itself.

48. (previously presented) The re-multiplexer module of claim 47, wherein said packet buffer further stores, for each stored packet, a Program Clock Reference Flag.

49. (previously presented) The re-multiplexer module of claim 36, wherein said packet identifier table further identifies some packets as having priority status such that, if said packet buffer is filled to a predetermined point, said input processor writes only packet having priority status to said packet buffer.

50. (previously presented) The re-multiplexer module of claim 49, wherein said predetermined point is half full.

51. (previously presented) A re-multiplexer module comprising:

an input processor controlled by a host processor in a packet processing system

a packet buffer; and

a packet identifier table wherein said input processor stores a packet in said packet buffer if said packet has an identifier listed in said packet identifier table;

wherein said input processor further comprising a Program Clock Reference (PCR) detector that flags packets containing PCR data.

52. (previously presented) The re-multiplexer module of claim 51, further comprising an output processor that reads packets from said packet buffer and selectively adds said packets to one of two or more output data streams.

53. (previously presented) The re-multiplexer module of claim 52, further comprising PCR correction circuitry in said output processor that extracts PCR data from said packets flagged as containing PCR data.

54. (previously presented) The re-multiplexer module of claim 51, wherein said table comprises:

an active table containing values used by the input processor to select packets for storage in said packet buffer; and

a pending table containing values that can be modified by the host processor while the active table is being used by the input processor,

55. (previously presented) The re-multiplexer module of claim 54, further including a switching mechanism allowing the host processor to make the pending table the active table, such that the active table becomes the pending table.

56. (previously presented) The re-multiplexer module of claim 52, further comprising a time reference generator, wherein said input processor associates a time stamp with each packet.

57. (previously presented) The re-multiplexer module of claim 56, wherein said output processor uses said time stamp to calculate an amount of time taken to pass a particular packet through said re-multiplexer module.

58. (previously presented) The re-multiplexer module of claim 57, wherein said output processor further comprises Program Clock Reference (PCR) correction circuitry that removes PCR jitter using said amount of time taken to pass a particular packet through said re-multiplexer module.

59. (previously presented) The re-multiplexer module of claim 51, wherein said packet buffer stores, for each stored packet, a Program Clock Reference Flag.

60. (new) The re-multiplexer module of claim 51, wherein PCR detector writes a said flag into said packet buffer in association with each flagged packet stored in said packet buffer.

61. (new) The re-multiplexer module of claim 51, wherein said PCR detector is configured to:

check for an adaptation field in each packet input to said input processor,

if an adaptation field is present in that packet, check a PCR flag bit in the adaptation field,

if said PCR flag bit is set, verify that PCR data in that packet is valid, and

write an additional flag indicating said packet contains valid PCR data.